

Association for Information Systems AIS Electronic Library (AISeL)

ACIS 2006 Proceedings

Australasian (ACIS)

2006

Document Variant Management – Facilitating Enterprise System Definition, Configuration, and Interoperability

Christian Janiesch

University of Münster, christian.janiesch@ercis.de

Stefan Seidel

University of Münster

Alexander Dreiling

SAP Australia Pty Ltd, alexander.dreiling@sap.com

Follow this and additional works at: <http://aisel.aisnet.org/acis2006>

Recommended Citation

Janiesch, Christian; Seidel, Stefan; and Dreiling, Alexander, "Document Variant Management – Facilitating Enterprise System Definition, Configuration, and Interoperability" (2006). *ACIS 2006 Proceedings*. 35.

<http://aisel.aisnet.org/acis2006/35>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2006 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Document Variant Management – Facilitating Enterprise System Definition, Configuration, and Interoperability

Christian Janiesch
Stefan Seidel
European Research Center for Information Systems
University of Münster
Münster, Germany
Email: christian.janiesch@ercis.de

Dr Alexander Dreiling
SAP Research
SAP Australia Pty Ltd
Brisbane, Queensland
Email: alexander.dreiling@sap.com

Abstract

Enterprise Systems (ES) provide a standardized means to assist operations and management. Being off-the-shelf software, ES provide a generic approach and they have to be configured to the enterprise's requirements. The customization of ES is a complex and costly task, therefore especially small and midsize enterprises (SME) flinch from doing so. Independent software vendors can assist SMEs in adapting ES to their needs. To be of reasonable economical value, it must be possible to facilitate the customization process with configuration mechanisms that allow generating customer specific processes and documents from a generic model-based repository. Several high-level configuration mechanisms are presented that add up to a comprehensive procedure for document variant management.

Keywords

Configuration, Document Engineering, Enterprise Systems, Interoperability, Variant Management

MOTIVATION

Enterprise Systems (ES) are typically understood as bundles of technology and business expertise that are used to support an organization's management and operations in a holistic way (Davenport 1998; Klaus et al. 2002; Ragowsky and Somers 2002). ES are provided as generic off-the-shelf solutions by ES vendors and implemented and maintained in significant collaborative projects by vendor, customer, and consultants, creating a close relationship between vendor and customer (Markus 2000).

One of the major problems with large-scale ES has been the fact that the generic character of the system and the specific character of the user organization requires for change. Typically, the system and organizational practices do not change in separation, but in combination. Davenport's (1998) assertion that the organization must change to fit into the system or even Markus' (2004) concept of "technochange," referring to the deliberate use of a system to change the organization are contrasted with a branch of literature that is concerned with making ES configuration more comprehensive (Becker et al. 2002; Dreiling et al. 2005; Soffer et al. 2005). The latter contributions aim at enabling a wider range of organizational actors to participate in ES configuration, reduce associated risks, and making it more feasible to adapt an ES within small and midsize enterprises (SMEs). In the SME sector it is not only important that configuration is made more comprehensive for customers, but also that vendors - or "channel partners" of vendors (compare SAP's (2005) SME strategy) - are enabled to quickly and easily compose ES that are preconfigured to a large extend for a small range of customers. This need is driven by the fact that SMEs typically define their competitive advantage through products or services rather than through more efficient administrative processes. SMEs expect such administrative processes to be supported by an ES and rely on "best practices" offered by ES vendors. However, processes related to procurement, sales, human resources, accounting, and the like vary in different industries and countries stressing the need for preconfigured solutions for smaller groups of customers.

The process of creating ES solutions for smaller customer groups and adapting these solutions intuitively by the customer in an economical way can be supported by model-based mechanisms. It is important that configuration is no longer exclusively feasible for large customer organizations and that configuration can also be intuitively and directly influenced by non-IT staff of customer and consulting organizations. This requires ES vendors to build more generic building blocks of ES functionality which can be reused and recontextualized in different contexts.

Componentized software engineering was already perceived as important nearly 40 years ago (McIlroy 1969) and is a well-developed discussion within computer science (Sametinger 1997). With the advent of ES based on the principles of Service-Oriented Architectures (SOA) (Erl 2005), ES construction based on ES components seems to become feasible.

We aim at contributing to this process by discussing an approach to business document variant management. Business documents are legally binding, mutual agreements that are exchanged between partners while conducting business. Due to the fact that such business documents are exchanged using different structural or exchange standards, with different technologies, and in different industries and countries, the need arises to adapt a generic business document to different contexts in order to avoid to define, e.g., a purchase order over and over again. We (a) propose a configuration process from the most generic business document to a specific one, (b) outline methods and technologies that can be used throughout this process, and (c) provide examples for our propositions.

This paper is structured as follows: Chapter 2 introduces the domain as well as the methodological foundations of this research. Chapter 3 elaborates on document variant management covering the configuration process and its methods. Chapter 4 provides a brief and illustrative example. The paper concludes with an evaluation and outlook to further research.

REQUIREMENTS ON BUSINESS DOCUMENTS AND CONFIGURATION

Business Documents State of the Art

Most state-of-the-art standards for document exchange have been developed for a specific industry or application (cf. e.g. CIDX, RosettaNet). Since this changes the working of the standards and has influence on the actual document's design, it is complicated to compare the standards let alone to let them communicate with each other.

One of the goals of new, emerging exchange standards is to avoid the confusion on how to treat data correctly in terms of structure, format, and meaning. The formers are currently achieved by utilizing XML as a markup language to describe and structure data. The latter is tried to achieve by introducing a further intermediate layer of business semantics in the communication process. The ebXML Core Components Technical Specification (CCTS) enables the definition of data types for document exchange as well as proposes a concept on how to organize, i.e. aggregate and associate, data on a higher level to create business documents (Crawford 2003). The general idea is to provide core component types that can be restricted to serve as a data type for basic core components which in turn can be aggregated to aggregate core components. The latter can be part of other aggregate core components as association core components. Every core component used in a certain context is called a Business Information Entity.

Being a meta standard CCTS does not provide any concrete implementation but simply a way of standardizing business semantics. The Universal Business Language (UBL) and the Open Applications Group Integration Specification (OAGIS), e.g., are two initiatives to provide a uniform language for business document exchange based on CCTS (Meadows and Seaburg 2004; Bosak and McGrath 2006; Open Applications Group 2006).

Apart from the CCTS-based standards mentioned above, a number of other, mostly industry-specific exchange standards exists that focus on static message definitions that have not enabled a comparable degree of flexibility or interoperability with other standards. Most of these approaches form their own data types which are usually based on XML data types and their own approach to document assembly. For the rest, EDIFACT is a mature standard for document interchange. It is widely used in Europe; its Northern American counterpart is ANSI ASC X12 (United Nations Economic Commission for Europe (UNECE) 2006). Their structure is similar. EDIFACT is not based on XML; it provides its own messaging structure as well as semantics. However, a structured composition of reusable components for messaging is not provided.

Summarizing, the development of current exchange standards is being influenced by other intermediate standards that facilitate communication and standardization of messages and documents. The first notable layer is XML as a general exchange markup language and more recently CCTS is implemented as a method of business semantics standardization. The acceptance of one general business language is still due but might be the long-term solution to standardizing the building blocks concept proposed by CCTS and facilitating the configuration of business documents.

Related Work on Configuration

In order to enable configurative information modeling, different approaches of model adaptation have been developed. Some approaches focus on "model transformation", as it is by the Model Driven Architecture (MDA) (Soley and OMG Staff Strategy Group 2000). Implementations of model transformation mechanisms can be found in the form of so-called Meta CASE Tools like the Generic Modeling Environment (GME) (Ledeczki et al. 2001; Agrawal et al. 2002) and Metaview (Findeisen 1994). Model transformation aims at generating a destination

model out of an original model, whereas the languages of both models can diverge extensively. Structural patterns are identified in the model of the initial modeling language via an algorithmic search and they are transformed into equivalent patterns of a model of the targeted modeling language. Transformations are performed by using transformation rules that are defined for each combination of the original and destination language (Engels et al. 1997). Model transformation approaches are characterized by a high universality of the operators used for the definition of transformation rules (e.g. *Create New, Replace, Same, Create Reference, Create Link, Delete, Refer else Create, Create inside, Refer to*) which makes high user competencies necessary. Transformations are usually employed in the software industry to adapt software to different operating systems or computer platforms. In the fields of process and document modeling, e.g., it is not necessarily required to transform models.

Other types of approaches focus on building views onto a model system. These views are then considered as perspectives which result from user requirements. Exemplary approaches of this type are the Semantic Object Model (SOM) (Ferstl and Sinz 1998), the Architecture of Integrated Information Systems (ARIS) (Scheer 2000), the Zachman Framework (Zachman 1987), the Open Systems Architecture for Computer Integrated Manufacturing (CIM-OSA) (ESPRIT Consortium AMICE 1998), MEMO (Frank 1994), and Viewpoints (Finkelstein et al. 1992). A common characteristic of these approaches is that the realization of multiple perspectives is restricted to providing different modeling views which result in different model types. The KOREAN framework (Becker et al. 2002; 2004) provides a generic framework that not only allows the configuration of perspectives by configuration of the model types but also by configuration of elements types as well as the actual elements of the model. Furthermore, it allows for synonym management and presentation variation to ease the understanding of the model. Soffer et al. (2005) describe configuration as an alignment process of adapting the enterprise system to the needs of the enterprise. Their approach allows for implementing process variants based on the values of certain attributes.

CONFIGURATION

Layers and Mechanisms of Configuration

The configuration mechanisms presented in the following provide means to directly or indirectly specify so-called contexts for the configuration of business documents masters. The following figure gives an overview of the general procedure.

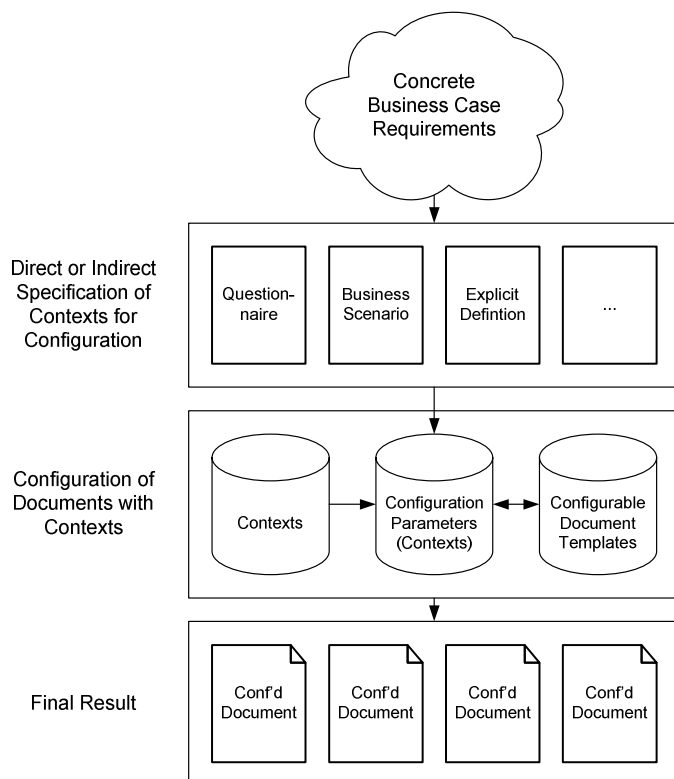


Figure 1: Layers of Configuration

A business document master is a template document comprising all possible characteristics. The actual structure for the requested document is derived from this master by means of model projection (cf. Becker et al. (2002; 2004) for more insights on the workings of model projection). Context drivers cluster contexts into distinct

groups. CCTS proposes eight distinct context drivers that can be utilized for the configuration of any business document (*process, product, industry, geopolitical, official constraint, role, supporting role, and system capability*) (Crawford 2003). All configurable documents are annotated with configuration parameters that provide information on how to configure the document given a certain context. In a future scenario, this contextualization of documents might be assisted by also adding contextual cues concerning the users of the system (Goh et al. 2004). The aim of providing different mechanisms rather than just allowing specifying the applicable contexts directly is to provide high-level mechanisms that can be used by others than IT staff. The result of each configuration procedure is always a set of contexts that is matched against those configuration parameters annotated at the documents. The aim is to guide the user through the configuration process rather than to confront him with all the options and all possible results at once since this leads to frustration rather than success (Roussinov and Chen 1999).

Note that the model-dependant configuration, especially the configuration via configuration points (CP), operate slightly different since they explicitly query available contexts or respectively do not necessarily make use of existing contexts at all. Naturally free modification does not abide to this or any other framework either.

The configuration is always twofold: In a first step appropriate documents and components are selected from the repository. In a second step these elements are configured according to the contextual parameters specified. To allow for a maximum of flexibility, manual configuration mechanisms should also be available to provide controlled as well as free adaptation. The combination of all these configuration mechanisms provides the developer with the means to retrieve the necessary documents from the repository (outer configuration) and in a second step re-use the specified contextual information to perform an inner configuration, i.e. to configure the content of each document and component. For the second step, it will most certainly not be necessary to redefine the parameters via questionnaires or business scenarios. However, a refinement based on all the contexts available and explicit configuration mechanisms are reasonable mechanisms to detail the requirements. This step typically includes a seamless transition to the more manual configuration mechanisms. Suggestions for high-level configuration mechanisms, which facilitate the configuration of business documents, include, e.g., questionnaires, business scenarios, context selection, mind map-based modeling, and the explicit use of CPs. Cf. the following figure for a clustering of the configuration mechanisms:

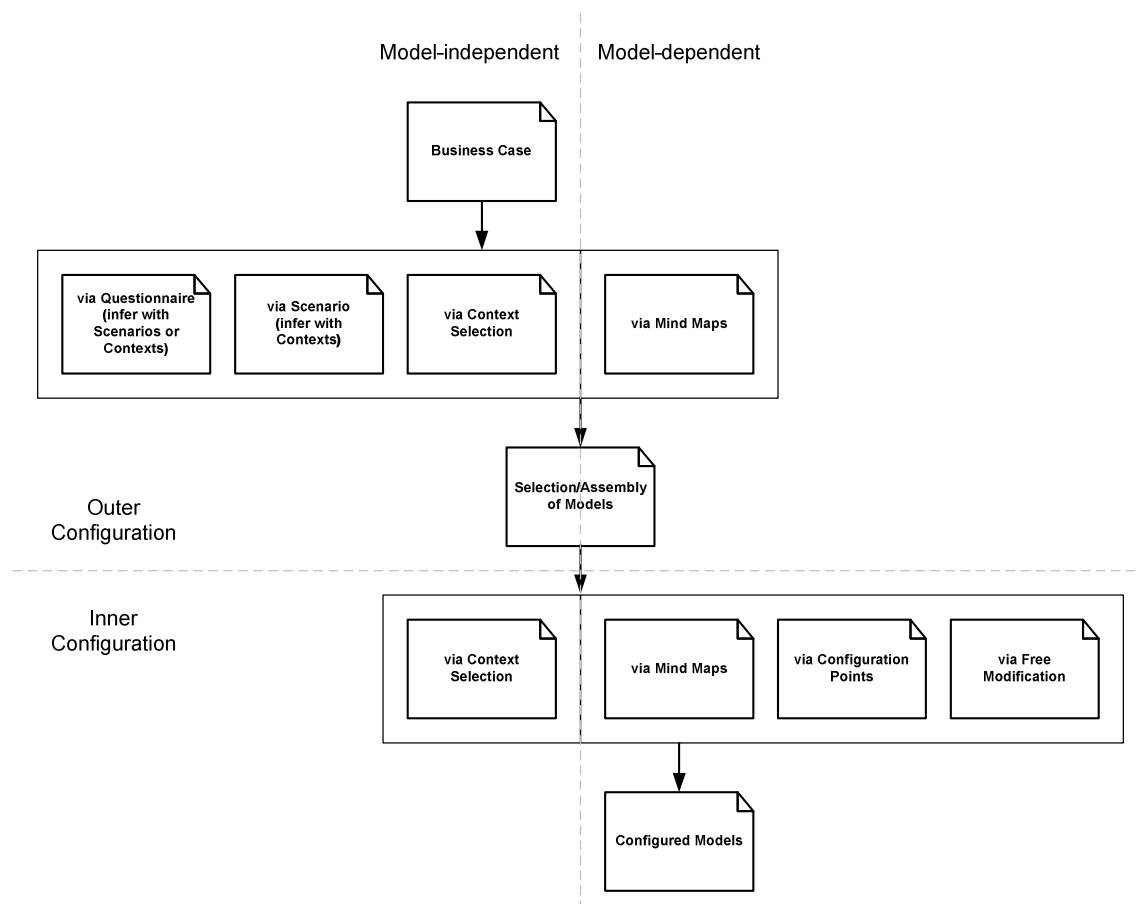


Figure 2: Clustering of Configuration Mechanisms

Model-independent Configuration

Starting from a business perspective the knowledge of or about specific business document is not necessarily available to the developers. Thus, it is more beneficial to describe the context in which the collaboration takes place than to directly search for a suitable model. This is only the next step which is computer-assisted by the prior definition of contexts to the document masters by domain and IT experts.

Questionnaires

Questionnaires are utilized to give the developer the possibility to specify requirements more or less from a business perspective and therefore model-independently. So the questionnaire is intended to be a high-level approach rather than being exhaustive and, thus, does not compare to existing check-box approaches like SAP's Implementation Guide (IMG). Within the questionnaire questions relevant to the situation of the enterprise and the business to be conducted are posed to position the enterprise and to provide initial input for the outer and inner configuration of the documents. The answers to the questions provide the system with contextual parameters that are then used to select the appropriate documents from the repository. In this case a contextual parameter may be a business scenario (see below) or the contexts themselves. Furthermore, it might be possible to employ an ontology-based approach that infers the contexts that are relevant to the specific implementation (Kim et al. 2005). The results of this configuration step lead to a rough selection of business documents and parameters to configure the documents. Cf. the following figure for an overview:

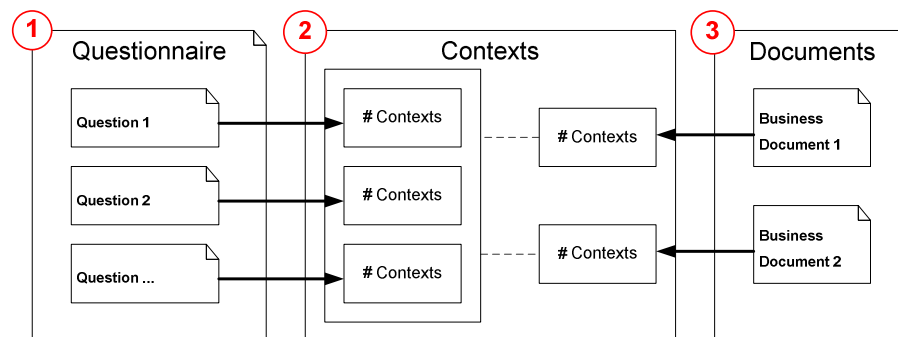


Figure 3: Questionnaire-based Configuration

Every answer to a question in a questionnaire (1) is linked to a number of contexts (2), e.g., if you produce the answer "Germany" to the question "Which country are you based in?" most certainly the geopolitical context "Germany" is linked to this possible answer. Document masters (3) and their components are annotated with a number of contexts (2). Thus, after answering the questionnaire the resulting set of contexts is matched against the contexts annotated at the documents in the repository to retrieve and configure the documents accordingly. Instead of directly annotating contexts as in the figure above (*# Contexts*), groups of contexts, e.g. scenarios, can be annotated also. Questionnaires are essentially an approach for outer configuration.

Business Scenarios

An alternative configuration mechanism for high-level configuration is the scenario-based or pattern-based configuration. By selecting one of the available business scenarios, which cover common business processes and actors, a pre-selection of semantically connected documents can be achieved (e.g. dispatch advice and receipt advice). A more detailed outer configuration can include product and industry specifics that provide further input for the inner configuration, too. The business scenarios, which are offered for configuration, group together documents mainly by the context drivers *process* and *role*. The results of questionnaire-based configuration lead to a rough selection of business documents. Cf. the following figure for an overview.

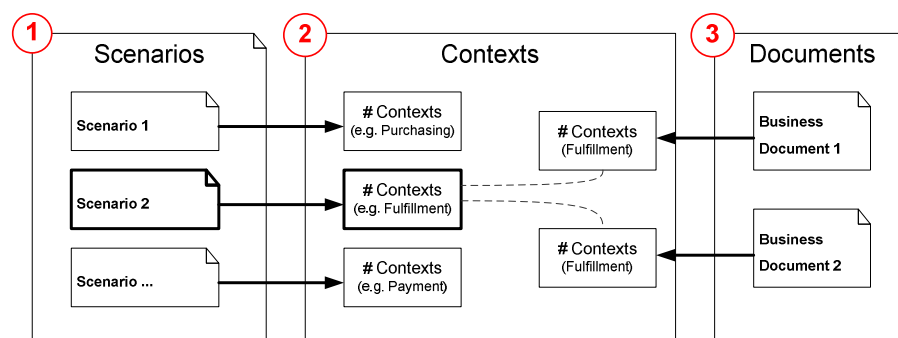


Figure 4: Scenario-based Configuration

The process is essentially the same as the above. Instead of answering questions a scenario is chosen directly from a repository (1). Each scenario is annotated with contexts as are document masters (3). Based on this information document masters are retrieved (2). Consistency of the scenarios can be achieved, e.g., by using an own code list for the process context driver that clusters documents according to the scenarios. Exemplarily in the figure above “Scenario 2” has been selected, thus, the context *Fulfillment* is of relevance. This is matched against the repository and two documents with the same context could be identified. This, too, is essentially an approach for outer configuration.

Context Selection

Contexts can also be specified explicitly as parameters for the outer and/or inner configuration. The coarse configuration mechanisms described above might not specify the requirements adequately enough so that a low-level approach has to be applied consecutively. A directory of all possible context drivers and contexts is presented that can be selected or deselected from. For consistency reasons this entails that contexts, which are mutually exclusive, must be highlighted to avoid confusion. Whenever contexts are selected or deselected and this leads to the fact that documents, which are currently applicable, or parts thereof are considered not useful anymore, a warning has to be displayed before they are omitted. In the first step, the results of this configuration lead to a granular selection of appropriate documents that, in a second step, can be configured according to the selected contexts. Cf. the following figure for an example.

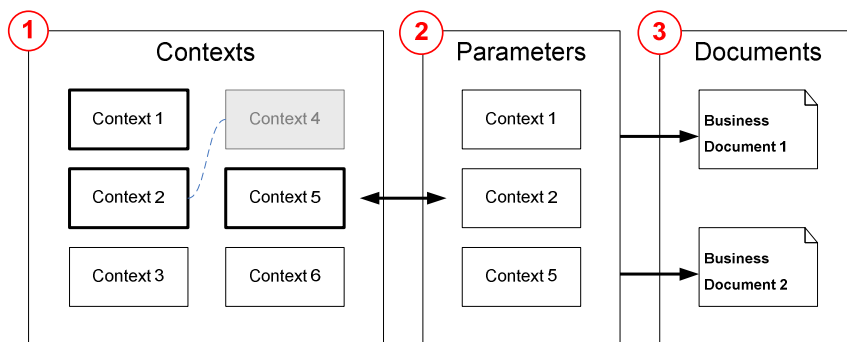


Figure 5: Configuration via Context Selection

Contexts 1, 2, and 5 have been selected. Context 4 is ruled out by Context 2 (1). This combination of contexts is matched against the annotated contexts of the business documents (2). Document 1 and 2 are the result (3). Context selection is an approach for outer as well as inner configuration.

Model-dependent Configuration

Instead of selecting contexts indirectly via a questionnaire or a business scenario, models are configured explicitly. So the focus of this approach is rather oppositional yet complementary since it allows the further sophistication of the model-independent specification on a detailed level.

Mind map-based Configuration

An interactive approach is the mind map-based modeling of business documents. Based on the requirements the system developer starts modeling a mind map (Buzan 2003) of the business document that is needed. Either in a two stepped approach or in an interactive way the document is finalized. The former requires the developer to model all the entities he reckons to be necessary and attaches them to the main topic. Then the model is checked against the repository for occurrences of the document respectively components. Applicable documents are ranked and offered for selection. This can be done by a comparison of subcomponents and data types (Crawford 2003), keywords and naming (cf. Yuan et al. 2004), ontologies (cf. Kim et al. 2005) and/or other meta-data (cf. Päivärinta et al. 2002). When selecting the document, the mind map is rearranged to fit to the structure of the document and components not initially included are added. All changes have to be validated by the modeler before proceeding. Furthermore, all components that do not match the selected document are highlighted. They can either be assigned to already existing components or can be matched against the repository so that the process starts over.

The latter approach provides components for instantiation of the modeled elements instantly while modeling and, thus, provides a means to model the document interactively. While the first approach supports modeling from scratch, the latter aims at standardizing the document by suggesting the course of modeling to the user. Cf. the following figure for an example.

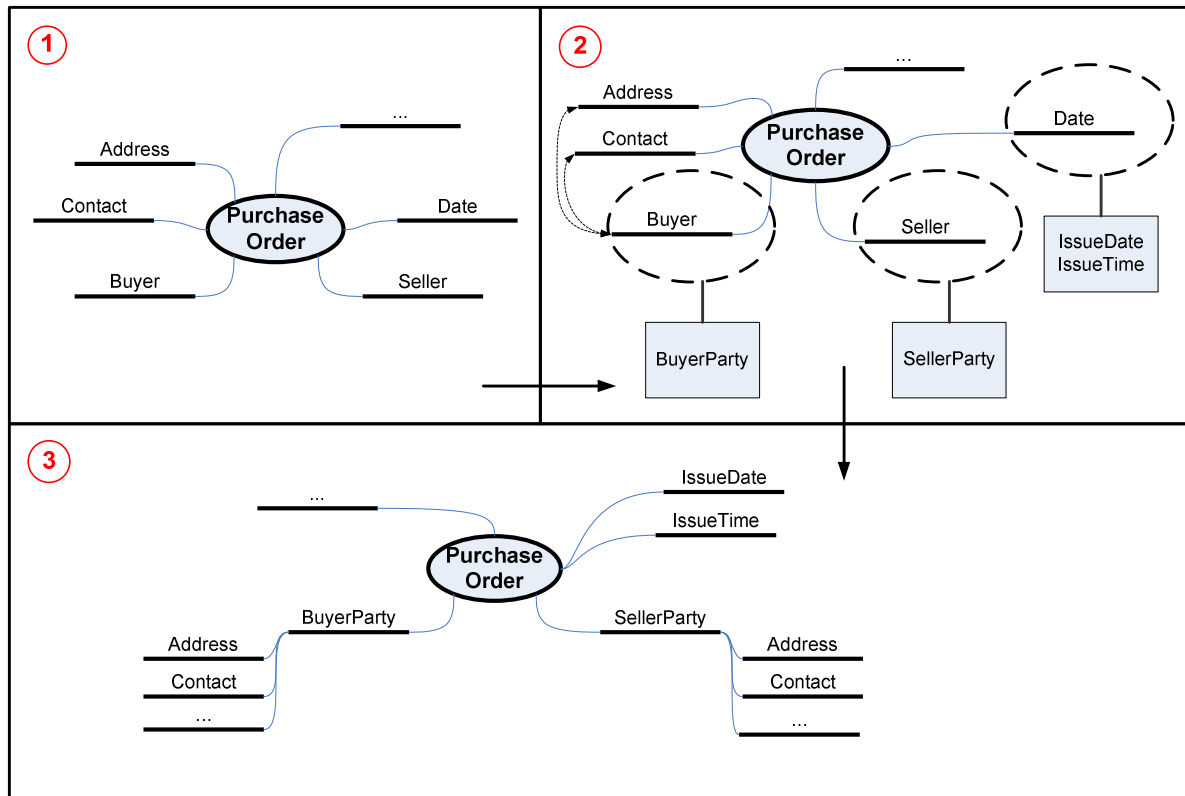


Figure 6: Mind map-based Document Specification

At first the document is modeled in a mind map that encompasses all the required components (*Address*, *Contact*, *Buyer*, *Seller*, *Date* and additional information subsumed as “...”) (1). In a second step, all components are compared with components available in the repository and suggestions are made on how to instantiate the elements (2). *Date* for example is proposed as either *IssueDate* or *IssueTime* or both and *Buyer* and *Seller* are changed to *BuyerParty* and *SellerParty* incorporating their substructure so that *Address* and *Contact* are rearranged. In a third step, these options are acknowledged and the document is assembled (3).

Configuration via Configuration Points (CP)

After selecting different contexts, the documents and components can be configured explicitly by selecting and/or configuring an element directly (for an extensive overview of possible manipulations cf. Becker et al. (2002; 2004)). So, configuration via CP is a supplementary approach to the configuration via contexts described above differing in the direction of configuration. While the former granular configuration mechanism of context selection answers the question “I select this context, what happens to the selection of business documents?” the latter answers the question “I want to change this object, what are my options?” Upon request, for every object the possible configuration parameters are displayed and can be selected from. This leads to a granular outer configuration which in a second step provides the necessary input for inner configuration. Cf. the following figure for a rough example.

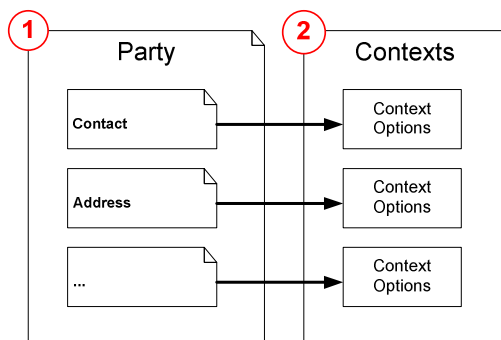


Figure 7: Configuration via Configuration Points

In a first step of configuration the component *Party* was found to be necessary (1). However its subcomponents *Contact* and *Address* had to be modified since, e.g., their cardinality was not appropriate. So, the annotated

contexts at each element are reviewed (2) to see if an adequate context is available. Since in a real world scenario, the pre-configured components might not be sufficiently flexible, the possibility of manual interference with the configuration must be allowed. This can be done in a controlled manner via the CPs. Instead of choosing a context, the constraint or omission can be entered manually. Concerning the figure above this would mean that in (2) not the annotated contexts and their implications are reviewed but a constraint would be manually entered, e.g. constraining each *Party* to one and only one *Contact* with one and only one *Address*.

Apart from this granular configuration of documents, shortcuts such as only “mandatory elements only” or “no multiplicity greater 1” could enhance the speed of configuration for complete document masters.

Free Modification

Anything goes. As a final resort the model can be twisted as needed. This, however, makes it an extremely specialized model that most certainly cannot be reused and is only applicable in a very special scenario. It is not recommended to do so, since maintenance and management of these models is exceedingly more complicated and costly.

EVALUATION AND FURTHER RESEARCH

Being off-the-shelf software, ES provide a generic approach and they have to be configured to the enterprises requirements. To be of reasonable economical value it must be possible to facilitate the customization process with configuration mechanisms that allow generating customer specific processes and documents from a generic model-based repository. A process and methods to configure documents have been proposed, justified and applied by means of a simple example.

To justify the conceptual “overhead” of configuration, its benefits have to be valued and contrasted with the drawbacks and limitations. Since no comprehensive empirical data exists on this subject, benefits and drawbacks can only be discussed argumentatively. While the examples above only hints at the benefits and is rather aimed at illustrating the procedure, the following facts prevail:

First and foremost the proposed approach is a top-down approach rather than the currently employed bottom-up approach. This entails some benefits that have not been realized before in practice. A top-down approach prohibits the proliferation of business documents which are created based on a perceived need during implementation. The decisions on what to implement are shifted up to a higher instance while the concrete implementation is deferred to the latest possible moment. The whole process is model driven and abstracts from implementation issues that unnecessarily complicate things in early development. Model driven development enables faster comprehension of the problem at hand and, thus, facilitates faster and easier adjustment to the task and produces results earlier. Inherent in the concept described above is component-orientation and re-use thereof allowing less redundancy in implementation.

Even though the approach offers many benefits, certain drawbacks and limitations are inevitable that have to be considered and weighed against the advantages in a real-world case.

The full configuration of any business document involves a lot of parameterization. Almost any element must be annotated with appropriate contexts from several code lists. This entails an enormous initial research effort on what to configure in which way. So it has to be ensured that the administration is economical and user-friendly (for further evidence on ease of use cf. Ma et al. (2001)). Furthermore, maintenance of the configuration parameters is complex. While high-level configuration is fairly straight forward, the inherent complexity of the necessary granular configuration of business documents is rather complicated. It is therefore very important that a simple access and management makes most issues transparent to the user. While the aim is to shorten the overall time of implementation, longer delays in implementation can be expected, when new elements are introduced that are subject to an approval process.

All in all, the contextual configuration of business documents provides a promising concept for the management of large heterogeneous amounts of business documents that have a common basis such as the repository of an ERP software vendor. Through context-based configuration mechanisms, the means of modifications are flexible and yet restrictive. Documents can be configured to very diverse requirements but only in a pre-defined and traceable way. This controlled flexibility offers a feasible concept for document variant management.

Further research and sophistication is needed in the definition of the actual context annotation, concerning consistent storage, and in specification of feedback drivers for a harmonization process. A means is needed to automate or at least to assist the harmonization of findings, which are based on reoccurring change, new contexts, and/or significant change that does not originate from an isolated case. This entails on a more general level: A framework for situational and evolutionary document management is needed.

REFERENCES

- Agrawal, A., Levendovszky, T., Sprinkle, J., Shi, F., Karsai, G. "Generative Programming via Graph Transformations in the Model-Driven Architecture," in *OOPSLA Workshop on Generative Techniques in the Context of Model Driven Architecture*, Seattle, 2002, pp. 1-11.
- Becker, J., Delfmann, P., Dreiling, A., Knackstedt, R., Kuropka, D. "Configurative Process Modeling - Outlining an Approach to Increased Business Process Model Usability," in *Proceedings of the 15th Information Resources Management Association Conference (IRMA 2004)*, New Orleans, LA, 2004, pp. 615-619.
- Becker, J., Delfmann, P., Knackstedt, R., Kuropka, D. "Konfigurative Referenzmodellierung," in *Wissensmanagement mit Referenzmodellen - Konzepte für die Anwendungssystem- und Organisationsgestaltung*, J. Becker, R. Knackstedt (eds.), Heidelberg, 2002, pp. 25-144.
- Bosak, J., McGrath, T. *Universal Business Language 2.0 Public Review Draft*, Organization for the Advancement of Structured Information Standards (OASIS), 2006, <http://docs.oasis-open.org/ubl/prd-UBL-2.0/> [accessed 2006-02-22].
- Buzan, T. *The Mind Map Book - Radiant Thinking - Major Evolution in Human Thought*, BBC Active, 2003.
- Crawford, C. *Core Components Technical Specification - Part 8 of the ebXML Framework*, United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), 2003, http://www.unece.org/cefact/ebxml/CCTS_V2-01_Final.pdf [accessed 2006-02-22].
- Davenport, T. "Putting the Enterprise into the Enterprise System," *Harvard Business Review* (76:4), 1998, pp. 121-131.
- Dreiling, A., Rosemann, M., Aalst, W. M. P., Sadiq, W., Khan, S. "Model-driven Process Configuration of Enterprise Systems," in *Proceedings of the Wirtschaftsinformatik 2005*, Bamberg, 2005, pp. 691-710.
- Engels, G., Heckel, R., Taentzer, G., Ehrig, H. "A View-Oriented Approach to System Modelling Based on Graph Transformation," in *Proceedings of the 6th European Software Engineering Conference (ESEC)*, Zürich, 1997, pp. 327-343.
- Erl, T. *Service-oriented Architecture - Concepts, Technology, and Design*, Prentice Hall, Englewood Cliffs, NJ, 2005.
- ESPRIT Consortium AMICE (ed.) *CIM-OSA - Open System Architecture for CIM*. Springer-Verlag, Berlin et al. 1998.
- Ferstl, O. K., Sinz, E. J. "SOM Modeling of Information Systems," in *Handbook on Architectures of Information Systems*, Vol I., P. Bernus, K. Mertins, G. Schmidt (eds.), Springer-Verlag, 1998, pp. 339-358.
- Findeisen, P. *The Metaview System*, Alberta, 1994.
- Finkelstein, A., Kramer, J., Nuseibeh, B., Finkelstein, L., Goedicke, M. "Viewpoints: a Framework for Integrating Multiple Perspectives in System Development," *International Journal of Software Engineering and Knowledge Engineering* (1:2), 1992, pp. 31-57.
- Frank, U. *Multiperspektivische Unternehmensmodellierung. Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung*, Habilitation, Oldenbourg, München, Wien, 1994.
- Goh, J.-M., Poo, D., Chang, K. "Incorporating Contextual Cues into Electronic Repositories," in *Proceedings of the 8th Pacific Asia Conference on Information Systems (PACIS 2004)*, Shanghai, 2004, pp. 472-484.
- Kim, S., Suh, E., Yoo, K., Lee, Y. "A Framework for Ontology-based Context Base Management System," in *Proceedings of the 9th Pacific Asia Conference on Information Systems (PACIS 2005)*, Bangkok, 2005, pp. 1202-1215.
- Klaus, H., Rosemann, M., Gable, G. G. "What is ERP?," *Information Systems Frontiers* (2:2), 2002, pp. 141-162.
- Ledeczi, A., Maroti, M., Bakay, A., Karsai, G., Garrett, J., Thomason, C., Nordstrom, G., Sprinkle, J., Volgyesi, P. "The Generic Modeling Environment," in *Workshop on Intelligent Signal Processing*, Budapest, 2001, pp. 19-25.
- Ma, J., Liang, J., Q., Z. "Component-based Approach to the Development of Internet EDI Software for Electronic Document Management," in *Proceedings of the 5th Pacific Asia Conference on Information Systems (PACIS 2001)*, Seoul, 2001, pp. 717-726.

- Markus, M. L. "Technochange Management: Using it to Drive Organizational Change," *Journal of Information Technology* (19:1), 2004, pp. 4-20.
- Markus, M. L. a. T., C. "The Enterprise System Experience - From Adoption to Success," in *Framing the Domains of IT Research - Glimpsing the Future Through the Past*, R. W. Zmud (ed.), Cincinnati, OH, Pinnaflex Educational Resources, 2000, pp. 173-207.
- McIlroy, M. D. "Mass Produced Software Components," in *Software Engineering*, P. Naur, B. Randell (eds.), Brussels, NATO Science Committee, NATO Scientific Affairs Division, 1969, pp. 138-150.
- Meadows, B., Seaburg, L. *Universal Business Language 1.0*, Organization for the Advancement of Structured Information Standards (OASIS), 2004, <http://docs.oasis-open.org/ubl/cd-UBL-1.0/> [accessed 2006-02-22].
- Open Applications Group, I. O. *Open Applications Group Integration Specification (OAGIS) Release 9.0*, Open Applications Group, Inc. (OAGi), 2006, <http://www.openapplications.org/downloads/oagidownloads.htm> [accessed 2006-02-22].
- Päivärinta, T., Tyrväinen, P., Ylimäki, T. "Defining Organizational Document Metadata: A Case Beyond Standards," in *Proceedings of the 10th European Conference of Information Systems (ECIS 2002)*, Gdansk, 2002, pp. 1154-1163.
- Ragowsky, A., Somers, T. M. "Special Section: Enterprise Resource Planning," *Journal of Management Information Systems* (19:1), 2002, pp. 11.
- Roussinov, D. G., Chen, H.-C. "Document Clustering for Electronic Meetings - An Experimental Comparison of Two Techniques," *Decision Support Systems* (27:1-2), 1999, pp. 67-79.
- Sameting, J. *Software Engineering with Reusable Components*, Springer, Heidelberg, 1997.
- SAP AG *SAP Solutions for Small and Midsize Enterprises: SAP® Business One and mySAP™ all-in-one*, SAP AG, Walldorf, 2005.
- Scheer, A.-W. *ARIS - Business Process Modeling*, 3rd Edition, Springer-Verlag, Berlin, 2000.
- Soffer, P., Golany, B., Dori, D. "ERP Modeling - A Comprehensive Approach," *Information Systems Frontiers* (28:6), 2005, pp. 673-690.
- Soley, R., OMG Staff Strategy Group *Model Driven Architecture (White Paper)*, Object Management Group, Needham, 2000.
- United Nations Economic Commission for Europe (UNECE) *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport*, United Nations Economic Commission for Europe (UNECE), 2006, <http://www.unece.org/trade/untidd/welcome.htm> [access 2006-02-22].
- Yuan, Z., Li, X.-B., Chen, J. "OBDMS - A Document Management System," in *Proceedings of the 8th Pacific Asia Conference on Information Systems (PACIS 2004)*, Shanghai, 2004, pp. 918-928.
- Zachman, J. A. "A Framework for Information Systems Architecture," *IBM Systems Journal* (26:3), 1987, pp. 277-293.

ACKNOWLEDGEMENTS

The work published in this paper is (partly) funded by the E.C. through the ATHENA IP. It does not represent the view of E.C. or the ATHENA consortium, and authors are solely responsible for the paper's content.

COPYRIGHT

Christian Janiesch, Alexander Dreiling, Stefan Seidel © 2006. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.